VPC (Virtual Private Cloud)

i)It’s like having your private space in the AWS cloud.

ii)you can think of it as your own virtual network where you can keep your data safe and separate from others.

iii) you get decide who can access your VPC and what they can do there.

iv)with VPC, you can create a network environment in the cloud that feels like a regular network setup.

Benefits of VPC

1)Isolation : your VPC is isolated from other people’s VPCs and the internet, giving you privacy and security.

2)Customization : you have full control over how your VPC is set up, including IP addresses, subnets, and security settings.

3)Security : you can setup security features like security groups and access control lists to control who can access your resources.

Subnet

i)A subnet is like a neighbourhood within you VPC.

ii)It helps organize your network into smaller, manageable parts.

iii)Subnet helps with things like organizing resources, using IP addresses efficiently, and enhancing security by isolating parts of the network.

iv)They're defined by a subnet mask, which determines the size of the subet and the range of IP addresses it includes.

v) A subnet range of Ip addresses in your VPC. A subnet must a single Avaliability zone. After you add subnet ,you can deploy the resources in your VPC.

Subnet Mask

i)it's a set of numbers that helps define the boundaries of a subnet .

ii)Subnet masks are used to determine which part of an IP address belongs to the network and which part belong to the host.

iii)They are written in special notation that helps routers and other devices understand how to route traffic within a network.

For eg. In the IP address "192.168.1.0/16" the "/16" indicates that the first 16 bits are reserved for network and 16 bits for host address.

This subnet has 65536 host addresses and remaining for network addresses.

Eg.

Subnet mask “255.0.0.0” is equivalent to “11111111.00000000.00000000.00000000” in binary.

Subnet mask “255.255.255.64” is equivalent to “11111111.11111111.11111111.01000000” in binary.

Route Table :

i)Route Table is like map of your network.

ii)It tells your data where to go within your network and how to get there.

iii)Route Tables are made up of rules(routes) that determine the path your network traffic should take.

Internet Gateway(IGW):

i)An Internet Gateway is like bridge between your VPC and the internet.

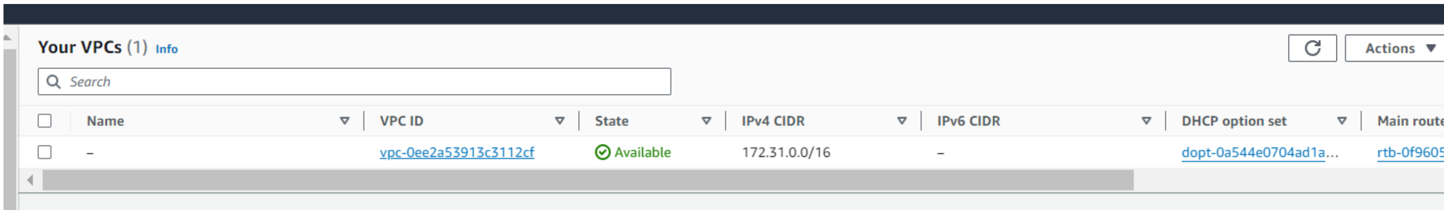
ii)It allows instances within your VPC to communicate with resources outside of the VPC and vice versa.

iii)It’s essential for accessing the internet from your VPC and for allowing internet traffic to reach your resources.

VPC helps you create your own private network in the cloud, complete with subnets for organization and security, route tables for directing traffic and Internet Gateway for connecting to the internet. It’s like building your own digital neighbourhood where you control everything.

Creating a VPC

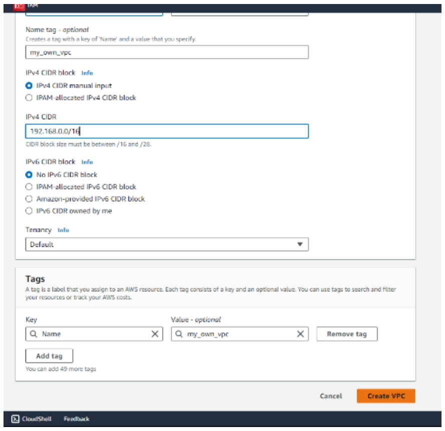
Step I : search VPC in services and click on it then click on create VPC .



Your VPCs (1) Info

□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route
□	–	vpc-0ee2a53913c3112cf	Available	172.31.0.0/16	–	dopt-0a544e0704ad1a...	rtb-0f9605

Step II : In VPC setting select VPC only ,give the name , IPv4 CIDR then click on create VPC.



Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
my_own_vpc

IPv4 CIDR block Info
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
192.168.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info
Default

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

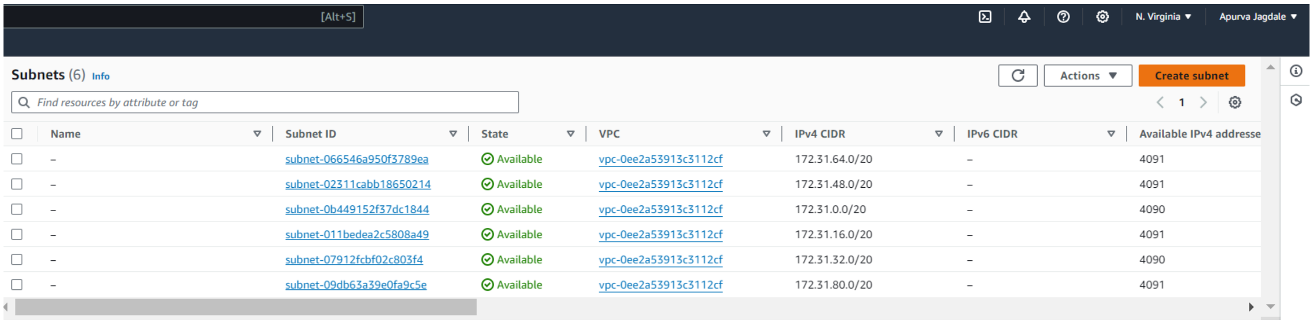
Key: Name
Value - *optional*: my_own_vpc

You can add 49 more tags

Step III : VPC is created Successfully.

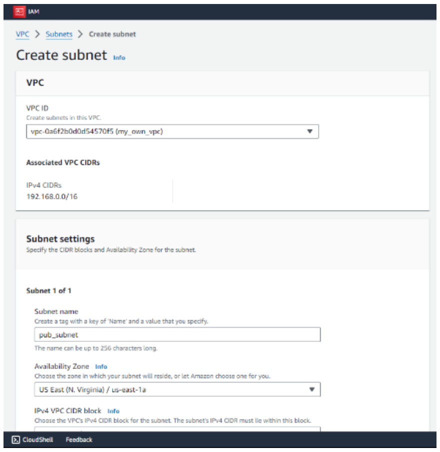
Creating a Subnet

Step I : In subnet, click on create subnet give VPC ID and give the name to public subnet(pub_subnet), select Availability zone, IPv4 CIDR give here and the click on create subnet.



Subnets (6) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresse
–	subnet-066546a950f3789ea	Available	vpc-0ee2a53913c3112cf	172.31.64.0/20	–	4091
–	subnet-02311cabb18650214	Available	vpc-0ee2a53913c3112cf	172.31.48.0/20	–	4091
–	subnet-0b449152f37dc1844	Available	vpc-0ee2a53913c3112cf	172.31.0.0/20	–	4090
–	subnet-011bedea2c5808a49	Available	vpc-0ee2a53913c3112cf	172.31.16.0/20	–	4091
–	subnet-07912fcbf02c803f4	Available	vpc-0ee2a53913c3112cf	172.31.32.0/20	–	4090
–	subnet-09db63a39e0fa9c5e	Available	vpc-0ee2a53913c3112cf	172.31.80.0/20	–	4091



VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0a6f2b0d0d54570f5 (my_own_vpc)

Associated VPC CIDRs

IPv4 CIDRs
192.168.0.0/16

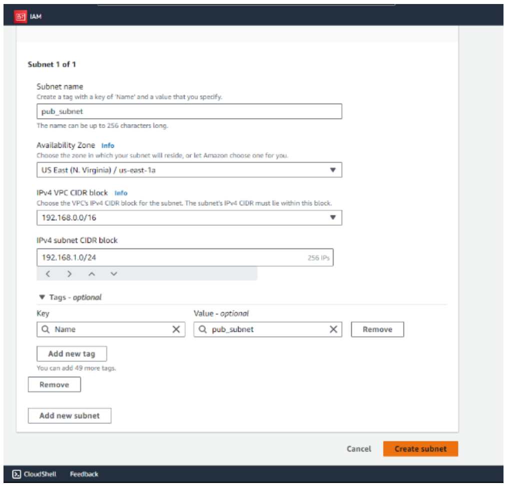
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pub_subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.



Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pub_subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/16

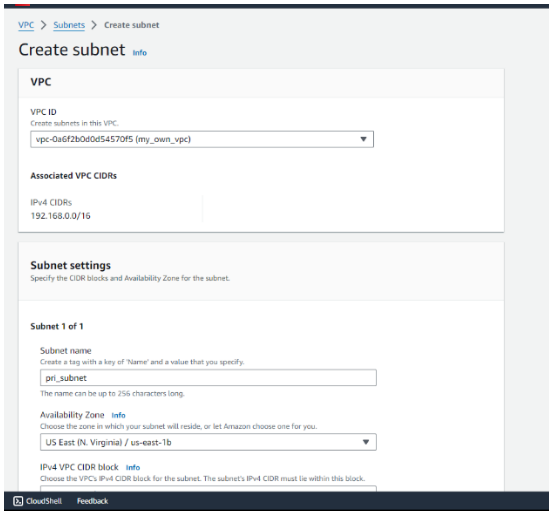
IPv4 subnet CIDR block
192.168.1.0/24 256 IPs

▼ Tags - *optional*

Key	Value - *optional*	
Name	pub_subnet	Remove

You can add 49 more tags.

Step II : now we creating private subnet ,follow same process.Give the VPC ID , name of private subnet(),select the Availability zone,give the IPv4 CIDR and click on create subnet.



VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0a6f2b0d0d54570f5 (my_own_vpc)

Associated VPC CIDRs

IPv4 CIDRs
192.168.0.0/16

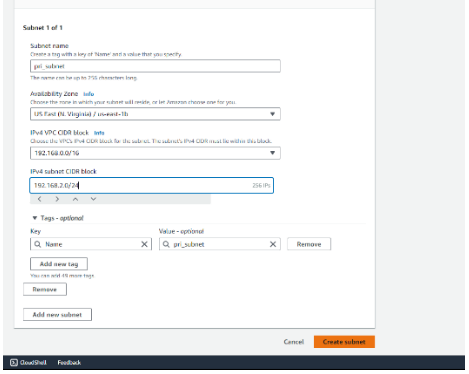
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pri_subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.



Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pri_subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b

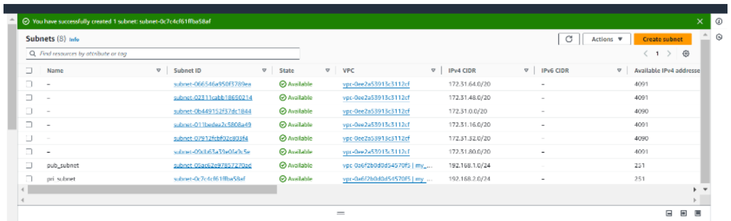
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/16

IPv4 subnet CIDR block
192.168.2.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*
Name	pri_subnet

You can add 49 more tags.



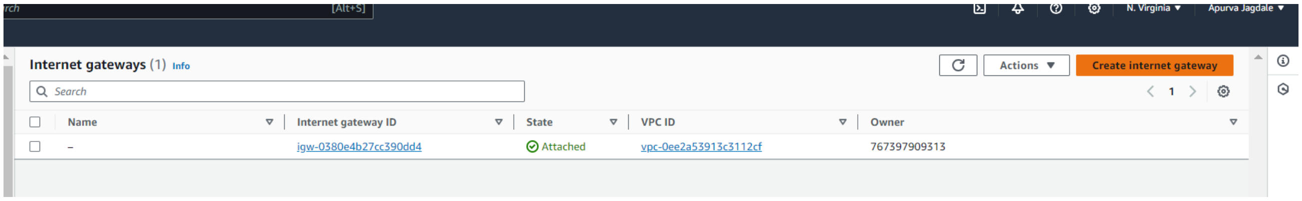
You have successfully created 1 subnet: subnet-0c7c4cf61ffba58af

Subnets (8) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresse
–	subnet-066546a950f3789ea	Available	vpc-0ee2a53913c3112cf	172.31.64.0/20	–	4091
–	subnet-02311cabb18650214	Available	vpc-0ee2a53913c3112cf	172.31.48.0/20	–	4091
–	subnet-0b449152f37dc1844	Available	vpc-0ee2a53913c3112cf	172.31.0.0/20	–	4090
–	subnet-011bedea2c5808a49	Available	vpc-0ee2a53913c3112cf	172.31.16.0/20	–	4091
–	subnet-07912fcbf02c803f4	Available	vpc-0ee2a53913c3112cf	172.31.32.0/20	–	4090
–	subnet-09db63a39e0fa0c5e	Available	vpc-0ee2a53913c3112cf	172.31.80.0/20	–	4091
pub_subnet	subnet-05ac62e97857270ad	Available	vpc-0a6f2b0d0d54570f5 | my_...	192.168.1.0/24	–	251
pri_subnet	subnet-0c7c4cf61ffba58af	Available	vpc-0a6f2b0d0d54570f5 | my_...	192.168.2.0/24	–	251

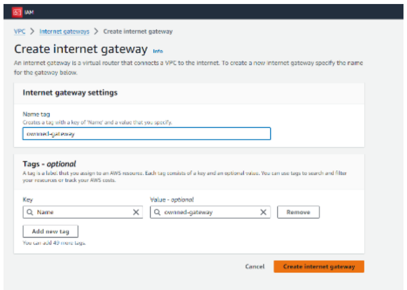
Creating the Internet Gateway

Step I : Click on Internet Gateway in VPC, give the name and click on create internet gateway.



Internet gateways (1) Info

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-0380e4b27cc390dd4	Attached	vpc-0ee2a53913c3112cf	767397909313



VPC > Internet gateways > Create internet gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag

Creates a tag with a key of 'Name' and a value that you specify.

ownned-gateway

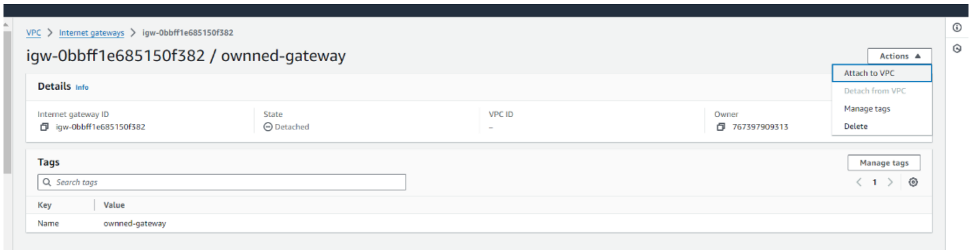
Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Name	ownned-gateway

You can add 49 more tags.

Step II : then click on Actions and then select Attach to VPC, select the created VPC and then click on attach internet gateway.



VPC > Internet gateways > igw-0bbff1e685150f382

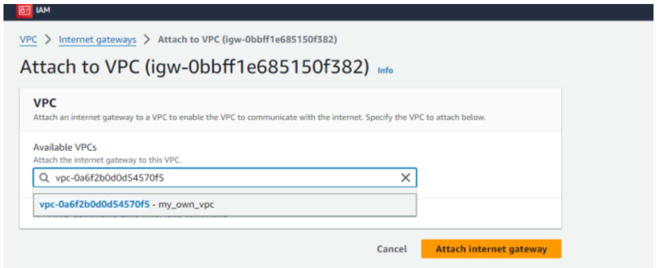
igw-0bbff1e685150f382 / ownned-gateway

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-0bbff1e685150f382	Detached	–	767397909313

Tags

Key	Value
Name	ownned-gateway



VPC > Internet gateways > Attach to VPC (igw-0bbff1e685150f382)

Attach to VPC (igw-0bbff1e685150f382) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

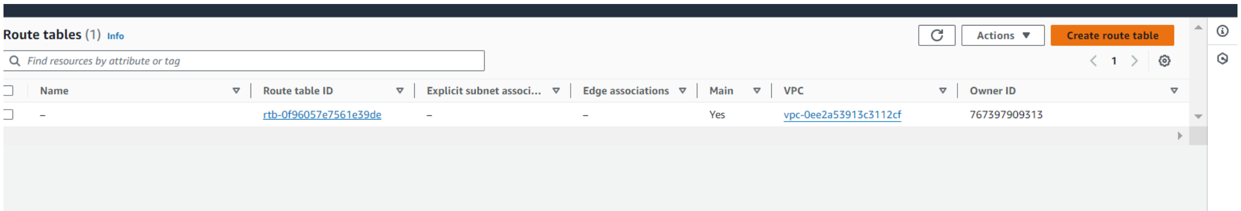
vpc-0a6f2b0d0d54570f5

vpc-0a6f2b0d0d54570f5 - my_own_vpc

Creating the Route Tables

FOR CREATING PUBLIC ROUTE TABLE

Step I : In VPC, Click on Route Tables and then click on create route table.

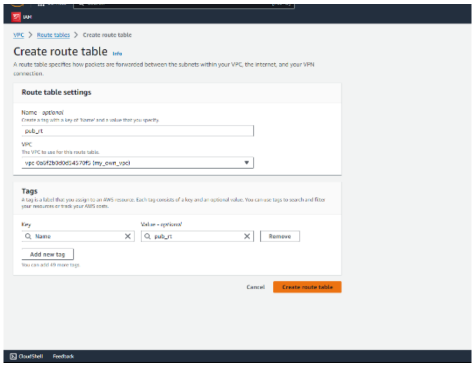


Route tables (1) Info

Find resources by attribute or tag

□	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
□	–	rtb-0f96057e7561e39de	–	–	Yes	vpc-0ee2a53913c3112cf	767397909313

Step II : Give name(pub_rt), select VPC and click on Create route table.



VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
pub_rt

VPC
The VPC to use for this route table.
vpc-0a6f2b0d0d54570f5 (my_own_vpc)

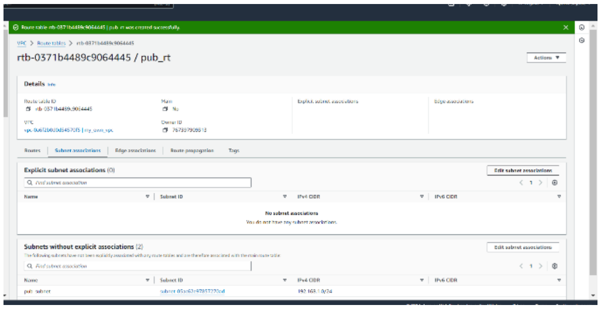
Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	pub_rt

You can add 49 more tags.

Step III : here pub_rt is created click on subnet association and edit this associate with public_subnet and click on save associations.



Route table rtb-0371b4489c9064445 | pub_rt was created successfully.

VPC > Route tables > rtb-0371b4489c9064445

rtb-0371b4489c9064445 / pub_rt

Details Info

Route table ID: rtb-0371b4489c9064445
Main: No
Explicit subnet associations
Edge associations
VPC: vpc-0a6f2b0d0d54570f5 | my_own_vpc
Owner ID: 767397909313

Explicit subnet associations (0)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR

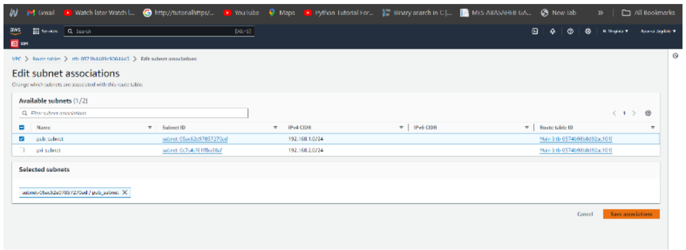
No subnet associations
You do not have any subnet associations.

Subnets without explicit associations (2)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
pub_subnet	subnet-05ac62c97857270ad	192.168.1.0/24	



VPC > Route tables > rtb-0371b4489c9064445 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

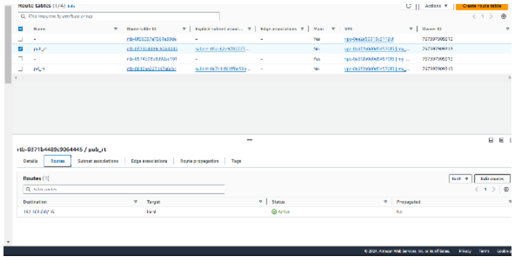
Filter subnet associations

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	pub_subnet	subnet-05ac62c97857270ad	192.168.1.0/24		Main (rtb-0574b98b8d92ac101)
□	pri_subnet	subnet-0c7c4c161fba58af	192.168.2.0/24		Main (rtb-0574b98b8d92ac101)

Selected subnets

subnet-05ac62c97857270ad / pub_subnet ✕

Step IV : then click on routes and click on edit routes in description select 0.0.0.0/0 it allowing all IP Addresses .



Route tables (1/4) Info

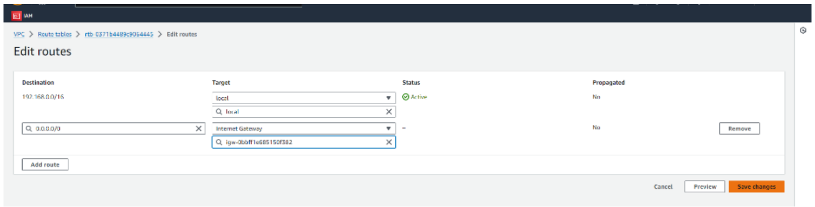
Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
☐ –	[illegible]	–	–	Yes	[illegible]	[illegible]
☑ pub_rt	[illegible]	[illegible]		No	[illegible]	[illegible]
☐ –	[illegible]	–	–	Yes	[illegible]	[illegible]
☐ pri_rt	[illegible]	[illegible]	–	No	[illegible]	[illegible]

rtb-0371b4489c9064445 / pub_rt

Routes (1)

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

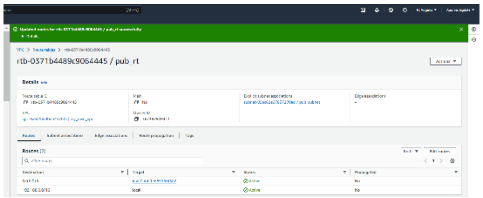
Step V : Add internet Gateway Target and click on save changes .



VPC > Route tables > rtb-0371b4489c9064445 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
192.168.0.0/16	local	Active	No	
0.0.0.0/0	Internet Gateway igw-0bbff1e685150f382	–	No	Remove



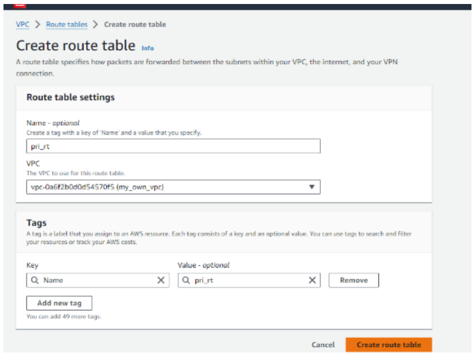
rtb-0371b4489c9064445 / pub_rt

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0bbff1e685150f382	Active	No
192.168.0.0/16	local	Active	No

FOR CREATING PRIVATE ROUTE TABLE

Step I : Again click on route table , give name then select the VPC and click on create Route table.



VPC > Route tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
pri_rt

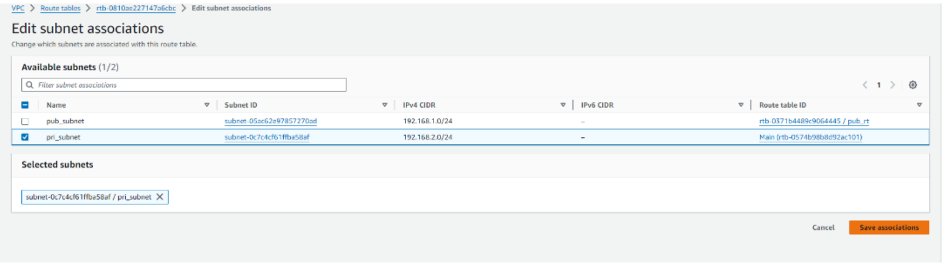
VPC
The VPC to use for this route table.
vpc-0a6f2b0d0d54570f5 (my_own_vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	pri_rt

You can add 49 more tags.

Step II : After creating ,in this select the subnet associations, then click to edit the subnet associations, select the private_subnet and click on save associations.


VPC > Route tables > rtb-0810ae227147a6cbc > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

Filter subnet associations
1

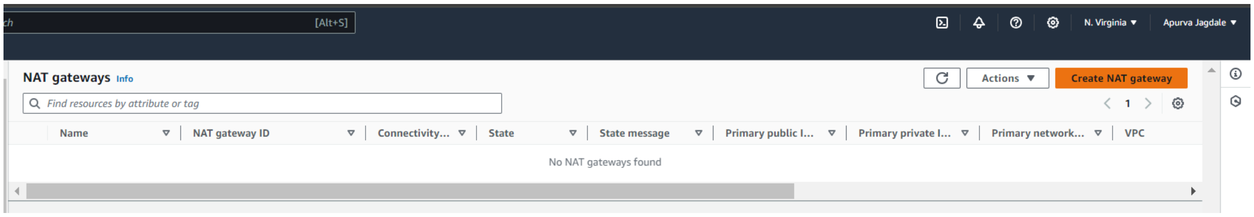
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
□	pub_subnet	subnet-05ac62e97857270ad	192.168.1.0/24	–	rtb-0371b4489c9064445 / pub_rt
☑	pri_subnet	subnet-0c7c4cf61ffba58af	192.168.2.0/24	–	Main (rtb-0574b98b8d92ac101)

Selected subnets

subnet-0c7c4cf61ffba58af / pri_subnet

Creating the NAT Gateways

Step I : In VPC, select the NAT Gateways then click on create NAT gateway.

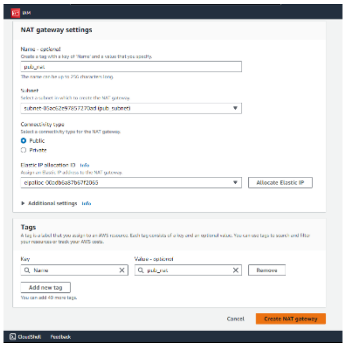


NAT gateways Info

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...	Primary network...	VPC

No NAT gateways found

Step II : Give the name to NAT gateway, select the public subnet, public connectivity type click on allocate Elastic IP and click on create NAT gateway.



NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
pub_nat
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-05ac62e97857270ad (pub_subnet)

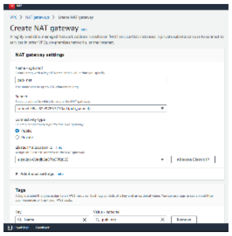
Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-00cdb6a87b67f2065

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Name	pub_nat

You can add 49 more tags.



VPC > NAT gateways > Create NAT gateway

Create NAT gateway Info

NAT gateway settings

Name - optional

pub-nat

Subnet

Connectivity type

● Public

○ Private

Elastic IP allocation ID Info

Allocate Elastic IP

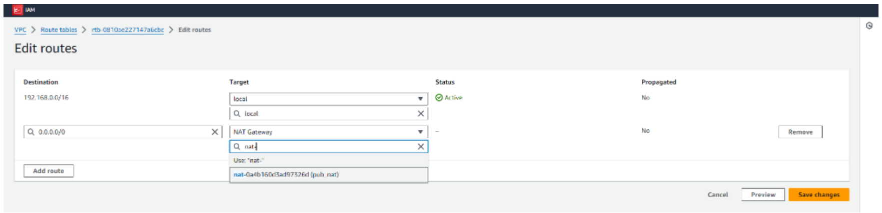
▶ Additional settings Info

Tags

Key: Name | Value - optional: pub-nat | Remove

ATTACHING THE NAT- GATEWAY TO PRIVATE – RT

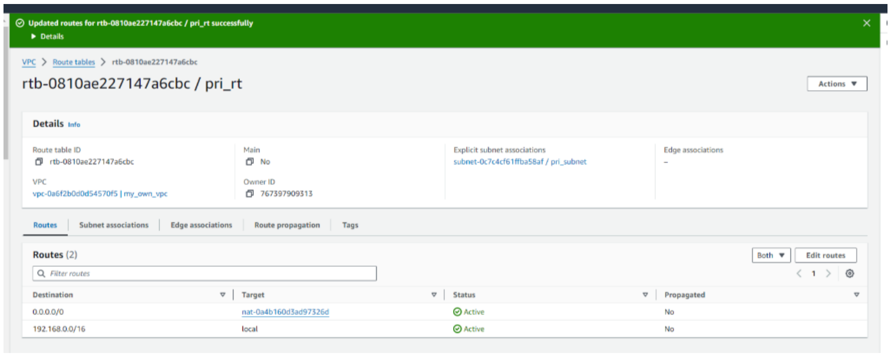
Step I : go to Route table, select the private_rt and click on Routes the edit routes select in description 0.0.0.0/0 and select NAT Gateway in Target and then click on save changes.



VPC > Route tables > rtb-0810ae227147a6cbc > Edit routes

Edit routes

Destination	Target	Status	Propagated	
192.168.0.0/16	local	Active	No	
0.0.0.0/0	NAT Gateway nat- Use: "nat-" nat-0a4b160d3ad97326d (pub_nat)	-	No	Remove



Updated routes for rtb-0810ae227147a6cbc / pri_rt successfully

▶ Details

VPC > Route tables > rtb-0810ae227147a6cbc

rtb-0810ae227147a6cbc / pri_rt

Details Info

Route table ID: rtb-0810ae227147a6cbc

Main: No

Explicit subnet associations: subnet-0c7c4cf61ffba58af / pri_subnet

Edge associations: –

VPC: vpc-0a6f2b0d0d54570f5 | my_own_vpc

Owner ID: 767397909313

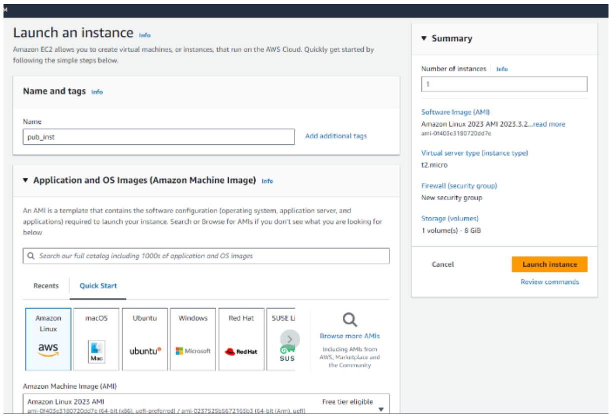
Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0a4b160d3ad97326d	Active	No
192.168.0.0/16	local	Active	No

Creating Instances in EC2

FOR CREATING PUBLIC INSTANCE

Step I : Go to EC2 service, click on Launch Instances, give the name it, select AMI by default (Amazon machine Image), select Instance type(t2.micro) , select key-pair.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: pub_inst

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible

ami-0f403e3180720dd7e (64-bit (x86), uefi-preferred) / ami-0237525b5672165b3 (64-bit (Arm), uefi)

Summary

Number of instances Info: 1

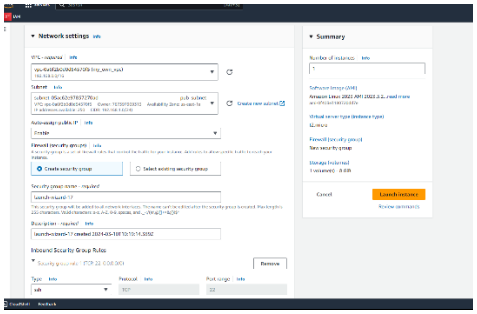
Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2...read more
ami-0f403e3180720dd7e

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Step II : In network settings, click on edit , add the created VPC , in subnet add pub_subnet, in auto assign Public IP enable it and launch it.



Network settings Info

VPC - required Info: vpc-0a6f2b0d0e545705 (my_own_vpc) 192.168.0.0/16

Subnet Info: subnet-05ac62e97857270ad pub_subnet
VPC: vpc-0a6f2b0d0e5457095 Owner: 707597000512 Availability Zone: us-east-1a
IP addresses available: 251 CIDR: 192.168.1.0/24

Auto-assign public IP Info: Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - required: launch-wizard-17

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - required Info: launch-wizard-17 created 2024-05-10T10:10:14.585Z

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type: ssh · Protocol: TCP · Port range: 22

Summary

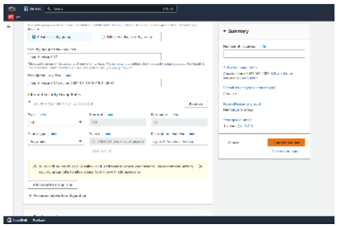
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2...read more

Virtual server type (instance type): t2.micro

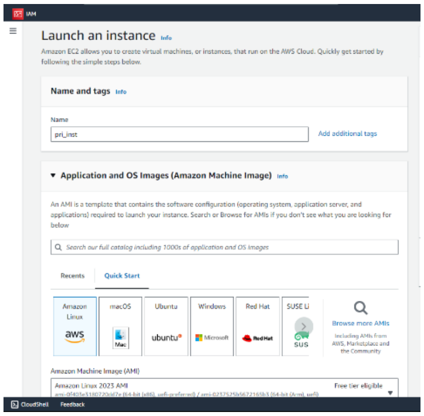
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB



FOR CREATING PRIVATE INSTANCE

Step I : click on launch Instances, give the name it, select the AMI and instance type and key-pair.



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: pri_inst

Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

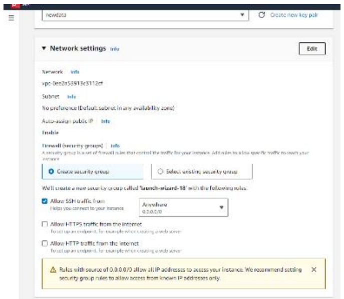
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible

ami-0f403e3180720dd7e (64-bit (x86), uefi-preferred) / ami-0237525b5672165b3 (64-bit (Arm), uefi)

Step II : In network settings, click on edit , select the created vpc and select the private subnet then in auto assign public IP, keep it disable as it is and launch it.



newdata

Network settings

Network: vpc-0ee2a53913c3112cf

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called 'launch-wizard-18' with the following rules:

☑ Allow SSH traffic from — Helps you connect to your instance — Anywhere 0.0.0.0/0

□ Allow HTTPS traffic from the internet — To set up an endpoint, for example when creating a web server

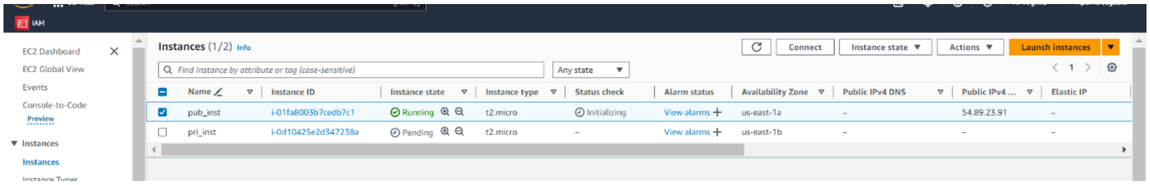
□ Allow HTTP traffic from the internet — To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Adding Security to Instances

FOR PRIVATE AND PUBLIC iNSTANCES

Step I : Select the Public instance then go to security in security groups .

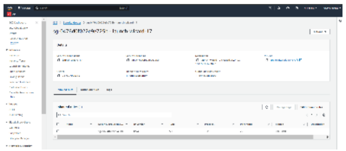


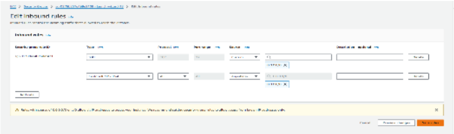
Instances (1/2)

Find Instance by attribute or tag (case-sensitive)

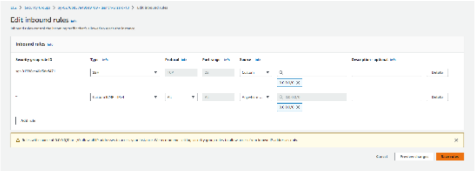
☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	pub_inst	i-01fa8003b7cedb7c1	Running	t2.micro	Initializing	View alarms +	us-east-1a	–	54.89.23.91	–
□	pri_inst	i-0d10425e2d347238a	Pending	t2.micro	–	View alarms +	us-east-1b	–	–	–

Step II : In Inbound Rules, click on edit and then add Custom ICMP – Ipv4 in protocol all,anywhere (0.0.0.0/0) and then save rules.



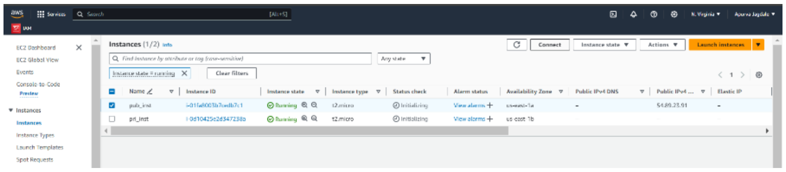


Follow same process for Private Instance.



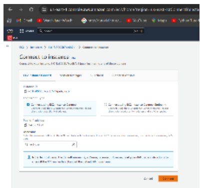
Connect the Instance

Step I : select the public Instance and connect it.

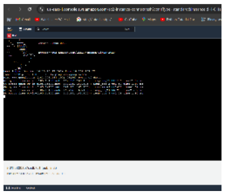


Instances (1/2)

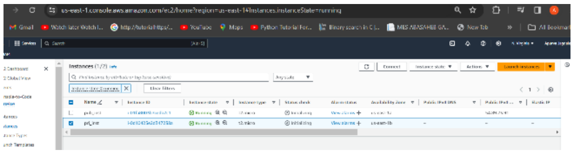
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑ pub_inst	i-01fa8003b7cedb7c1	Running	t2.micro	Initializing	View alarms +	us-east-1a	–	54.89.23.91	–
□ pri_Inst	i-0d10425c2d347238a	Running	t2.micro	Initializing	View alarms +	us-east-1b			

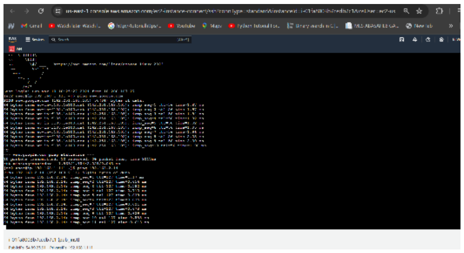


Step II : then ping the www.google.com to verify that we can access the internet or not.



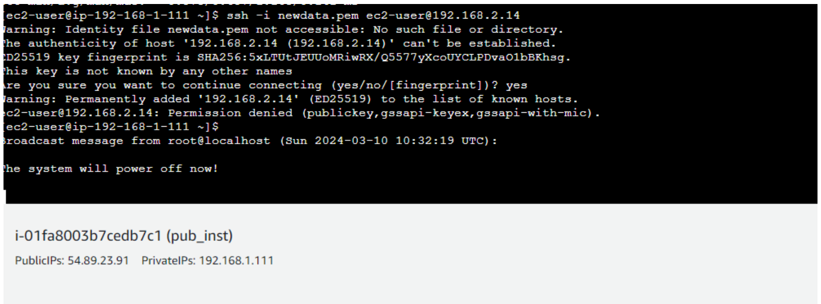
Step III : Then go to instances and copy the private instance of private IP and ping IP .





Step IV : it’s working fine, and we are able to ping without any packet loss .

Step V : to establish the connection and gain the access to another machine we can use “ssh -i newdata.pem ec2-user@ip”



```
ec2-user@ip-192-168-1-111 ~]$ ssh -i newdata.pem ec2-user@192.168.2.14
Jarning: Identity file newdata.pem not accessible: No such file or directory.
The authenticity of host '192.168.2.14 (192.168.2.14)' can't be established.
ED25519 key fingerprint is SHA256:5xLTUtJEUUoMRiwRX/Q5577yXcoUYCLPDvaO1bBKhsg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Jarning: Permanently added '192.168.2.14' (ED25519) to the list of known hosts.
ec2-user@192.168.2.14: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
ec2-user@ip-192-168-1-111 ~]$
Broadcast message from root@localhost (Sun 2024-03-10 10:32:19 UTC):

The system will power off now!
```

i-01fa8003b7cedb7c1 (pub_inst)

PublicIPs: 54.89.23.91 PrivateIPs: 192.168.1.111